

## CLAIMS

1. A method for managing an object in memory, comprising:
  - assigning the object to an assigned frame wherein the object can be released when the assigned frame is released;
  - 5 detecting an attempt to place a reference to the object in an older frame, the older frame being older than the assigned frame; and
  - reassigning the object to a reassignment frame that is at least as old as the older frame.
2. A method for managing an object in memory as recited in Claim 1, wherein the  
10 reassignment frame is the older frame.
3. A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object.
4. A method for managing an object in memory as recited in Claim 1, wherein  
15 assigning the object to the assigned frame comprises associating a frame identifier with a reference of the object.
5. A method for managing an object in memory as recited in Claim 1, wherein assigning the object to the assigned frame comprises associating a frame identifier with the object and detecting an attempt to place a reference to the object in an older frame is  
20 performed using the frame identifier.
6. A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises

comparing a first frame identifier associated with the object with a second frame identifier associated with the older frame.

7. A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises

5 comparing a first address associated with the object with a second address associated with the older frame.

8. A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory.

10 9. A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory by examining a distinguishing bit or a distinguishing set of bits.

10. A method for managing an object in memory as recited in Claim 1, wherein  
15 detecting an attempt to place a reference to the object in an older frame comprises determining whether the object is in stack memory or heap memory; and heap memory is uniquely identified by a heap frame identifier.

11. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame that is at least as old as the older frame  
20 comprises recursively detecting whether the object references any younger object.

12. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame that is at least as old as the older frame

comprises recursively detecting whether the object references any younger object and reassigning any referenced younger object to the reassignment frame.

13. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises resetting a first frame identifier  
5 associated with the object to be the same as a second frame identifier associated with the reassignment frame.

14. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame.

10 15. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and storing overflow in an overflow area associated with the reassignment frame.

15 16. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises expanding the reassignment frame and moving the object into the reassignment frame.

17. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating a reference to the object.

20 18. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame comprises moving the object into the reassignment frame and updating all references to the object.

19. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object to a reassignment frame is assisted by a display.
20. A method for managing an object in memory as recited in Claim 1, wherein reassigning the object includes tracing reassigned space.
- 5 21. A method for managing an object in memory as recited in Claim 1, further comprises modifying an allocation site of the object.
22. A method for managing an object in memory as recited in Claim 1, further comprises modifying a frame creation site.
23. A method for managing an object in memory as recited in Claim 1, further  
10 comprises learning reassignment information.
24. A method for managing an object in memory as recited in Claim 1, further comprising performing thread-local garbage collection.
25. A method for managing an object in memory as recited in Claim 1, further comprising storing call path information associated with an allocation site of the object.
- 15 26. A method for managing an object in memory as recited in Claim 1, wherein detecting an attempt to place a reference to the object in an older frame is performed with hardware assist.
27. A computer program product for managing an object in memory, the computer program product being embodied in a computer readable medium and comprising  
20 computer instructions for:
- assigning the object to an assigned frame wherein the object can be released when the assigned frame is released;

detecting an attempt to place a reference to the object in an older frame,  
the older frame being older than the assigned frame;

reassigning the object to a reassignment frame that is at least as old as the  
older frame.

5     28.     A system for managing an object, comprising:

          a processor configured to:

              assign the object to an assigned frame wherein the object can be  
              released when the assigned frame is released;

10           detect an attempt to place a reference to the object in an older  
          frame, the older frame being older than the assigned frame;

              reassign the object to a reassignment frame that is at least as old as  
              the older frame; and

          a memory coupled to the processor, configured to provide the processor  
with instructions.

15     29.     A method for improving performance of a computer program, comprising:

          gathering a first set of escape data;

          providing a first compiled program using the first set of escape data;

          gathering a second set of escape data based on the first compiled program;

          and

20           providing a second compiled program using the second set of escape data;

          wherein the second compiled program is more optimized than the first  
compiled program.

30. A method for improving performance of a computer program as recited in Claim 29, wherein the second compiled program includes a greater amount of inlining than the first compiled program.
31. A method for improving performance of a computer program as recited in Claim 29, further comprising performing escape analysis on the second compiled program.